A Lecture Series on DATA COMPRESSION

Wavelets and Subband Coding

Abdou Youssef (Speaker)

Anastase Nakassis (MDVTG Manager)

Wavelet-Based Approximation (Introduction and Motivation)

Some "Desirables" in Approximation Theory

- **Building Block Property**: Availability of a single approximating function ϕ, called *a scaling function*, to serve as a building block for approximations (or samplings) of any given function $x(t)$
  - The approximation is done by having different translates of ϕ fitted as closely as possible to $x(t)$
  - The translates of ϕ are $(\phi_k)_k$, where $\phi_k(t) = \phi(t - k)$
  - The approximation is represented by the multiplier coefficients $(a_k)_k$ of the translates $(\phi_k)_k$
  - Mathematically, $x(t) \equiv \Sigma_k a_k \phi_k(t)$

- **Malleability Property**: Dilatability of the building block ϕ to yield higher- or lower-resolution approximations
  - A *dilate* of ϕ at scale j is $\phi_{j,0} = 2^{\frac{j}{2}}\phi(2^j t)$
  - The translates of dilates are $\phi_{j,k} = 2^{\frac{j}{2}}\phi(2^j t - k)$
  - Note that the domain size of $\phi_{1,k}$ is half that of ϕ, and the domain size of $\phi_{2,k}$ is quarter that of ϕ, and so on. Thus the music terminology "scale"
  - The multiplier $2^{\frac{j}{2}}$ is to to make the dilates have the same energy (i.e., square integral) as ϕ
  - The higher the scale j, the higher the resolution, because the domain size of the building block is finer

Some "Desirables" in Approximation Theory (Cont.)

- **Analysis/Decomposition Property (part 1)**: Ability to derive the lower-resolution coefficients from the next higher-resolution coefficients <u>without</u> any reference to the original function $x(t)$

- **Analysis/Decomposition Property (part 2)**: Ability to derive the coefficients of the error $e(t)$ (residual or difference) between a higher-resolution approximation and a lower-resolution approximation,
  - using only the higher-resolution coefficients
  - <u>without</u> any reference to the original function $x(t)$
  - using a building block ψ, called *wavelet*
  - mathematically, $e(t) = \Sigma_k b_k \psi_k$

- **Synthesis Property**: Ability to derive the higher-resolution coefficients directly <u>without</u> any reference to the original function $x(t)$

- **Good-fit Property**: Choice of a good building block ϕ for the application signal(s) $x(t)$ so that the error between the $x(t)$ and the approximations is as small as possible

Illustrations of Translates and Dilates

Mathematical Formulations

- Denote by $x_j(t)$ the approximation of $x(t)$ at scale j. That is,
  - $x_j(t) = \Sigma_k \, x_k \phi_{j,k}(t)$ for some $(x_k)_k$
  - $x_j(t)$ is the best approximation of $x(t)$ among all linear combinations of $(\phi_{j,k}(t))_k$

- Let $e_{j-1}(t) = x_j(t) - x_{j-1}(t)$ be the error between scale-j and scale-$(j-1)$ approximations of $x(t)$;

$$x_j(t) = x_{j-1} + e_{j-1}(t)$$

- Notation:

$$\begin{aligned} x_j(t) &= \Sigma_k \, x_k \phi_{j,k}(t) \\ x_{j-1}(t) &= \Sigma_k \, u_k \phi_{j-1,k}(t) \\ e_{j-1}(t) &= \Sigma_k \, v_k \psi_{j-1,k}(t) \end{aligned}$$

- Desirable properties
  - For every $x(t)$ in L^2, $x_j(t) \longrightarrow x(t)$ as $j \longrightarrow \infty$ (convergence in L^2)
  - Analysis property: derivability of $(u_k)_k$ and $(v_k)_k$ from $(x_k)_k$
  - Synthesis property: derivability of $(x_k)_k$ from both $(u_k)_k$ and $(v_k)_k$

Conditions for Achieving the Analysis and Synthesis Properties

- Choose the scaling function ϕ to be composable from its own higher-resolution dilates & translates

- Choose the wavelet ψ to be composable from the higher-resolution dilates & translates of ϕ

- Finally, higher-resolution dilates-&-translates of ϕ should be decomposable into lower-resolution dilates-&-translates of ϕ and ψ

- Mathematically
  - $\phi(t) = \Sigma_n \, p_n \phi(2t - n)$, for some sequence $(p_n)_n$
  - $\psi(t) = \Sigma_n \, q_n \phi(2t - n)$, for some sequence $(q_n)_n$
  - $\phi(2t - k) = \frac{1}{2} \Sigma_n \left[g_{2n-k} \phi(t - n) + h_{2n-k} \psi(t - n) \right]$, for some sequences $(g_n)_n$ and $(h_n)_n$

Relation to Subband Coding

- **Theorem**:
  Let $x_j(t) = \Sigma_k \, x_k \phi_{j,k}(t)$
  $x_{j-1}(t) = \Sigma_k \, u_k \phi_{j-1,k}(t)$
  $e_{j-1}(t) = \Sigma_k \, v_k \psi_{j-1,k}(t)$
  where
  $\phi(t) = \Sigma_n \, p_n \phi(2t - n)$
  $\psi(t) = \Sigma_n \, q_n \phi(2t - n)$
  $\phi(2t - k) = \frac{1}{2} \Sigma_n \, [g_{2n-k} \phi(t - n) + h_{2n-k} \psi(t - n)].$

  Then $(x_k)_k$ is related with $(u_k)_k$ and $(v_k)_k$ by the following subband coder:

Proof of the Theorem
(Analysis stage: from $(x_k)_k$ to $(u_k)_k$ and $(v_k)_k$)

- $\phi(2t-k) = \frac{1}{2}\Sigma_n \left[g_{2n-k}\phi(t-n) + h_{2n-k}\psi(t-n)\right]$

- $$\begin{aligned}\phi_{j,k}(t) &= 2^{\frac{j}{2}}\phi(2^j t-k) \\ &= 2^{\frac{j}{2}-1}\Sigma_n\left[g_{2n-k}\phi(2^{j-1}t-n) + h_{2n-k}\psi(2^{j-1}t-n)\right]\end{aligned}$$

- $\phi_{j,k}(t) = \frac{1}{\sqrt{2}}\Sigma_n\left[g_{2n-k}\phi_{j-1,n}(t) + h_{2n-k}\psi_{j-1,n}(t)\right]$

- $$\begin{aligned}x_j(t) &= \Sigma_k x_k\phi_{j,k}(t) \\ &= \Sigma_n\left(\Sigma_k \frac{g_{2n-k}}{\sqrt{2}}x_k\right)\phi_{j-1,n}(t) + \Sigma_n\left(\Sigma_k \frac{h_{2n-k}}{\sqrt{2}}x_k\right)\psi_{j-1,n}(t)\end{aligned}$$

- On the other hand,
  $x_j(t) = x_{j-1}(t) + e_{j-1}(t) = \Sigma_n u_n\phi_{j-1,n}(t) + \Sigma_n v_n\psi_{j-1,n}(t)$

- Therefore,
  - $u_n = \Sigma_n \frac{g_{2n-k}}{\sqrt{2}}x_k = \overline{u_{2n}}$
  - $v_n = \Sigma_n \frac{h_{2n-k}}{\sqrt{2}}x_k = \overline{v_{2n}}$

- That is, $(u_n)_n$ is the down-sampled $\frac{g}{\sqrt{2}}$-filtered $(x_k)_k$, and $(v_n)_n$ is the down-sampled $\frac{h}{\sqrt{2}}$-filtered $(x_k)_k$

Proof of the Theorem
(Synthesis stage: from $(u_k)_k$ and $(v_k)_k$ to $(x_k)_k$)

- $\phi(t) = \Sigma_n\, p_n \phi(2t - n)$

- $$\begin{aligned}\phi_{j-1,k}(t) &= 2^{\frac{j-1}{2}} \phi(2^{j-1}t - k)\\ &= 2^{\frac{j-1}{2}} \Sigma_n\, p_n \phi(2(2^{j-1}t - k) - n)\\ &= \Sigma_n\, \tfrac{p_n}{\sqrt{2}} 2^{\frac{j}{2}} \phi(2^j t - 2k - n)\\ &= \Sigma_n\, \tfrac{p_n}{\sqrt{2}} \phi_{j,2k+n}(t)\end{aligned}$$

- Similarly, $\psi_{j-1,n} = \Sigma_n\, \frac{q_n}{\sqrt{2}} \phi_{j,2k+n}(t)$

- $$\begin{aligned}x_j(t) &= x_{j-1}(t) + e_{j-1}(t)\\ &= \Sigma_k \left[u_k \phi_{j-1,k}(t) + v_k \psi_{j-1,k}(t)\right]\\ &= \Sigma_k \left[u_k\, \Sigma_n\, \tfrac{p_n}{\sqrt{2}} \phi_{j,2k+n}(t) + v_k\, \Sigma_n\, \tfrac{q_n}{\sqrt{2}} \phi_{j,2k+n}(t)\right]\\ &= \Sigma_k\, \Sigma_n \left[\tfrac{p_n}{\sqrt{2}} u_k + \tfrac{q_n}{\sqrt{2}} v_k\right] \phi_{j,2k+n}(t)\\ &= \Sigma_r \left\{\Sigma_k \left[\tfrac{p_{r-2k}}{\sqrt{2}} u_k + \tfrac{q_{r-2k}}{\sqrt{2}} v_k\right]\right\} \phi_{j,r}(t)\end{aligned}$$

  (where $r = 2k + n$)

- Since $x_j(t) = \Sigma_r\, x_r \phi_{j,r}(t)$, it follows that

  $$x_r = \sum_k \left[\frac{p_{r-2k}}{\sqrt{2}} u_k + \frac{q_{r-2k}}{\sqrt{2}} v_k\right]$$

  which is precisely the sum of the $\frac{p}{\sqrt{2}}$-filtered upsampled $(u_k)_k$ and the $\frac{q}{\sqrt{2}}$-filtered upsampled $(v_k)_k$

Illustration of Wavelets' Dynamic Adjustment to Regional Variations without Blockiness (Comparison with Whole DCT)

Mathematical Method for Computing the Four Filters $(g_n)_n$, $(h_n)_n$, $(p_n)_n$, $(q_n)_n$ (Symmetric Filters)

1. Define the z-transforms of the four filters, with a slight scale modification:
   $G(z) = \frac{1}{2}\Sigma_k\, g_k z^k$, $P(z) = \frac{1}{2}\Sigma_k\, p_k z^k$,
   $H(z) = \frac{1}{2}\Sigma_k\, h_k z^k$, $Q(z) = \frac{1}{2}\Sigma_k\, q_k z^k$

2. The perfect reconstruction condition (seen before):
   - $PR1$: $G(z)P(z) + H(z)Q(z) = 1$
   - $PR2$: $G(-z)P(z) + H(-z)Q(z) = 0$

3. Take $H(z) = -z^{-1}P(-z)$ and $Q(z) = -zG(-z)$, i.e.,
   $$h_k = (-1)^k p_{k+1} \;\; and \;\; q_k = (-1)^k g_{k-1}$$

4. That choice of H and Q satisfies PR2 and makes PR1 equivalent to
   $$PR'1: \; G(z)P(z) + G(-z)P(-z) = 1$$

5. **Theorem**: The symmetry of the filters along with $PR'1$ implies that for any $z = e^{-i\omega}$

$$P(z) = e^{-i\frac{m}{2}\omega} \cos^l(\tfrac{\omega}{2}) S(\cos\omega)$$
$$G(z) = e^{i\frac{m}{2}\omega} \cos^{\hat{l}}(\tfrac{\omega}{2}) \hat{S}(\cos\omega)$$

for some integers m, l and $\hat{l}$, and some polynomials S and $\hat{S}$, such that m, l and $\hat{l}$ have the same parity, and l and $\hat{l}$ are positive.

6. Let $N = \frac{l+\hat{l}}{2}$

7. Therefore,
   - $G(z)P(z) = (\cos^2(\frac{\omega}{2}))^N S(\cos\omega)\hat{S}(\cos\omega)$
   - $G(-z)P(-z) = (\sin^2(\frac{\omega}{2}))^N S(-\cos\omega)\hat{S}(-\cos\omega)$, because $-z = e^{-(\omega+\pi)}$

8. By letting $x = \sin^2(\frac{\omega}{2})$ and defining the polynomial $F(x) = S(\cos\omega)\hat{S}(\cos\omega) = S(1-2x)\hat{S}(1-2x)$, one concludes from the previous step and $PR'1$ the following equation

$$(1-x)^N F(x) + x^N F(1-x) = 1$$

9. The general solution of that equation is of the form

$$F(x) = R(x) + x^N T_0(x)$$

where $R(x)$ is a polynomial of degree $N-1$ satisfying

$$(1-x)^N R(x) + x^N R(1-x) = 1,$$

and $T_0(x)$ is an arbitrary polynomial such that

$$T_0(1-x) = -T_0(x)$$

10. The equation of $R(x)$ implies that

$$R(x) + x^N (1-x)^{-N} R(1-x) = (1-x)^{-N}.$$

Since $(1-x)^{-N} = \Sigma_{k\geq 0} \begin{pmatrix} N-k+1 \\ k \end{pmatrix} x^k$ and $R(x)$ is of degree $N-1$, it follows that

$$R(x) = \sum_{k=0}^{N-1} \begin{pmatrix} N-k+1 \\ k \end{pmatrix} x^k$$

11. Therefore,

$$F(x) = \sum_{k=0}^{N-1} \begin{pmatrix} N-k+1 \\ k \end{pmatrix} x^k + x^N T(\cos\omega)$$

where $T(\cos\omega) = T_0(x)$ is an arbitrary odd polynomial.

12. In conclusion, to get $S(\cos\omega)$ and $\hat{S}(\cos\omega)$, first factor $F(x)$ by means of root finding, then give some factors to $S(\cos\omega)$ and the remaining factors to $\hat{S}(\cos\omega)$.

Algorithm for Computing the Taps of the Filters

1. Input: specify l, $\hat{l}$, and the polynomial T
2. $N = \frac{l+\hat{l}}{2}$ and $F(x) = \Sigma_{k=0}^{N-1} \begin{pmatrix} N-k+1 \\ k \end{pmatrix} x^k + x^N T(1-2x)$
3. Find the roots of $F(x)$
4. Thus F is factored into $F = F_1 F_2 ... F_r$,
   where every F_k is a linear or quadratic polynomial in x
5. Input: specify the index set L of the factors going to S
6. $S(\cos\omega) := \Pi_{k\in L} F_k(x) = \Pi_{k\in L} F_k(-\frac{1-z}{2}^2 z^{-1})$
7. $\hat{S}(\cos\omega) := \Pi_{k\in \overline{L}} F_k(x) = \Pi_{k\in \overline{L}} F_k(-\frac{1-z}{2}^2 z^{-1})$
8. Compute the coefficients of
   $P(z) = z^{\frac{m-l}{2}} \left(\frac{1+z}{2}\right)^l \Pi_{k\in L} F_k(-\frac{1-z}{2}^2 z^{-1})$
9. Let $p_k =$ the coefficient of z^k in $P(z)$, for all k
10. Compute the coefficients of
    $G(z) = (-1)^{\hat{l}} z^{-\frac{m+\hat{l}}{2}} \left(\frac{1-z}{2}\right)^{\hat{l}} \Pi_{k\in \overline{L}} F_k(-\frac{1-z}{2}^2 z^{-1})$
11. Let $g_k =$ the coefficient of z^k in $G(z)$, for all k
12. Normalize $(p_k)_k$ and $(g_k)_k$ so that $\Sigma_k\, p_k = 2$ and $\Sigma_k\, g_k = 2$
13. Compute $h_k = (-1)^k p_{k+1}$ and $q_k = (-1)^k g_{k-1}$

Symmetric Filters Generation

- The Compression Algorithms Group has developed an engine that generates all symmetric filters, and plots their frequency response as well as their corresponding scaling functions and wavelets

Examples of Four-Filter Sets

Daubechies Orthogonal Wavelets

- In orthogonal wavelets the following holds:
  - $g_k = p_k$
  - $h_k = (-1)^k p_{k+1}$
  - $q_k = (-1)^k g_{k-1}$

- Thus, one filter fully specifies all the four filters

- Daubechies Orthogonal wavelets are the most popular

- The Compression Algorithms Group has developed an engine that generates all Daubechies filters, and plots their frequency response as well as their corresponding scaling functions and wavelets

Examples of Daubechies Wavelets and Filters

Some Research Research Topics

- Lossless Compression
  - Multistage Compression
  - Novel Predictive-based Compression
  - Selective Compression
  - "Interframe" Compression
  - Symbolic Coding
- Lossy Compression: Wavelets
  - Optimal Ways to Apply Wavelets for Compression
  - Best-Wavelet Selection
  - Multi-Wavelet compression
  - 3D Wavelet Compression (for Video)
- Statistical Modeling of Classes of Images for better Compression
- Error Resiliency
  - Error Protection (with Error Correcting Coding)
  - Error Propagation and Self-Synchronizing Coding
  - Coding with Unequal Error-Protection
- Image Quality
  - Metrics and Benchmark Tests

  - Use of Contrast-Sensitivity Functions for Dynamic Adjustment of the Compression Ratio to tailor it to the specific user/monitor/application

- Other Wavelet Applications: Zooming, Alignment, and Modeling

- Effect of Compression-based Information Loss on the Accuracy of Image Processing Algorithms